Towards Adversarial Resilience in Proactive Detection of Botnet Domain Names by using MTD

Christian Dietz*†, Gabi Dreo*, Anna Sperotto†, Aiko Pras†

* Research Institute CODE
Bundeswehr University Munich
Neubiberg, Germany
Email:{Christian.Dietz, Gabi.Dreo}@unibw.de

†Design and Analysis of Communication Systems
University of Twente
Enschede, The Netherlands
Email:{C.Dietz, A.Sperotto, A.Pras}@utwente.nl

***Abstract*—Artificial Intelligence is often part of state-of-the-art Intrusion Detection Systems. However, attackers use Artificial Intelligence to improve their attacks and circumvent IDS systems. Botnets use artificial intelligence to improve their Domain Name Generation Algorithms. Botnets pose a serious threat to networks that are connected to the Internet and are an enabler for many cyber-criminal activities (e.g., DDoS attacks, banking fraud and cyber-espionage) and cause substantial economic damage. To circumvent detection and prevent takedown actions, bot-masters use DGAs to create, maintain and hide C&C infrastructures. Furthermore, botmasters often release its source code to prevent detection, leading to numerous similar botnets that are created and maintained by different botmasters. As these botnets are based on nearly the same source code basis, they often share similar observable behavior. Current work on detection of DGAs is often based on applying machine learning techniques, as they are capable to generalize and to also detect yet unknown derivatives of a known botnets. However, these machine learning based classifiers can be circumvented by applying adversarial learning techniques. As a consequence, there is a need for resilience against adversarial learning in current Intrusion Detection Systems. In our work, we focus on adversarial learning in DNS based IDSs from the perspective of a network operator. Further, we present our concept to make existing and future machine learning based IDSs more resilient against adversarial learning attacks by applying multi-level Moving Target Defense strategies.**

***Index Terms*—Adversarial Learning, Resilience, DGA, Botnet, Proactive Detection**

I. Introduction

The Internet and many large-scale corporate Networks become increasingly managed by using artificial Intelligence (AI) [1]. AI is often part of state-of-the-art Intrusion Detection System (IDS). However, attackers use AI to improve their attacks and circumvent IDS systems, which is also referred to as adversarial learning. Bot-masters use adversarial learning strategies for improving Domain Name Generation Algorithms (DGAs) to make their Command and Control (C&C) infrastructures more resilient against take downs. Furthermore, botmasters often release its source code to prevent detection, leading to numerous similar botnets that are created and maintained by different botmasters. As these botnets are based on nearly the same source code basis, they share similar observable behavior, such as using the same DGA [2], [3]. Current related work on the detection of such DGAs is often based on applying machine learning approaches. One of the main reasons to use machine learning for the detection of DGAs is that machine learning based classifiers can abstract from the actual training set and detect new derivatives of the previously trained botnet DGAs. Thus, they are more resilient to small modifications than signature based classifiers. Numerous machine learning and deep learning based approaches have been proposed for the detection of botnets and their DGAs. However, the use of machine learning and deep learning is becoming more popular on the attackers side, which is commonly referred to as adversarial learning. Usually, adversarial learning is intended to learn a neural network based model that is able generate a modified input that is able to trick another existing classification system. Thus, these systems are usually referred to as generative adversarial networks (GAN) [4]. As a consequence, there is a need for adversarial resilience of detection systems. Some related work [5]–[7] already addresses this issue. However, all known approaches are only addressing very specific types of adversarial manipulation to existing classifiers. We discovered that a flexible concept that increases the resilience of existing and future classifiers throughout the whole training and deployment life-cycle is still missing. To the best of our knowledge, our work is the first that applies a multi-level resilience strategy based on Moving Target Defense (MTD) principles against adversarial learning attacks in botnet detection.

In this paper, we make the following contributions: (i) we have provided a brief state-of-the art analysis of machine learning based detection and adversarial learning approaches in the network security domain. (ii) we describe our concept to increase the robustness of existing and future machine learning based attack detection and prediction.

II. Attacker model and Assumptions

In this Section, we first describe the adversarial attacker types and the assumptions on which we based our work.

A. *Attack types:*

Adversarial learning based attacks are categorized into the three types, which are (i) black-box attacks, (ii) white-box attacks, and (iii) grey-box attacks. For all attack types, we

978-1-7281-4973-8/20/$31.00 © 2020 IEEE

assume that the attacker able obtain feedback from the classifier, i.e. if the attacker tries to resolve a generated domain name it either receives an feedback that the domain was resolved successfully or receives an error message that it was blocked/unresolvable.

a) White-box attacks: In white-box attacks, the attacker has (nearly) full knowledge of the internals of the attacked classifier. This includes the source code on which the classifier is based on, the confidence rates, (at least partially) the training data, the underlying architecture (Configuration of neural network layers) and the type of machine learning approach (e.g. LSTM vs Random Forest).

b) Black-box attacks: The black-box attacks are carried out with only the ability to issue requests and receive feedback accordingly. The attacker can make an educated guess about the type of machine learning classifier that would be suitable and therefore most likely used.

c) Grey-box attacks: In grey-box attacks, the degree of knowledge is somewhere between the white-box and the black-box attacks. In such a scenario an attacker, might have the ability to resolve generated domain names, but additionally, knows publicly available DGA algorithms that were probably used to train the classifier. Therefore, the attacker could achieve at least a full coverage of the training data set that was used for the targeted classifier.

B. Assumptions:

We assume that *attackers have limited resources* and time, as in a business model for botnets with a pay off [8]. Further, we assume that classifiers are based on *supervised learning* approaches, which means that labelled training and evaluation data is used to iteratively improve the classification model. With respect to the model architecture, we assume *stackable classifier models*, e.g. deep neural networks, where a neural network consists of multiple networks that are joined into a larger model that learns to effectively combine the sub-models. This is important to enable the interchange of sub-models that then form an ensemble classifier.

III. Concept

In this section, we present our concept by first introducing the challenges faced by a DNS based IDS to withstand adversarial attacks. Secondly, we introduce the requirements that our novel adversarial robust detection concept has to fulfill. Finally, we describe the components and process model of our novel concept in detail.

A. Challenges

The 4 main challenges to make a detection concept resilient against adversarial learning based attack mechanisms are:

a) Complexity: is a challenge due to the multiple stages of randomization of the overall system configuration and the consequently increased management overhead, that has to remain manageable for the network operators.

b) Reproducibility: is a challenge as we aim to combine MTD strategies and adversarial learning in our robust detection approach and both approaches add random changes to either our data sets or our overall system configuration. However, miss-classifications should be reproducible especially to enable improvement of the overall system.

c) Time: poses a challenge as the system configuration should change regularly in short periods of time, but training multiple models and an ensemble can cost a lot of time depending on the available hardware.

d) State quality: is challenging to be guaranteed and track in complex MTD systems, as they involve many random reconfigurations, which increase the risk of unexpected interference, between different configurations.

B. Requirements

We derived 5 requirements for our concept from [9] and [10].

a) Accuracy: is one of the key requirements of any classification system. The detection accuracy of adversarial resilient classifiers shall not be significantly decreased in comparison to an existing classifier.

b) Transparency/ Reproducibility: should ensure and enable human operators to track configurations. This is especially important, in case of increased numbers of miss-classifications of the system. The operator should be able to reproduce the exact configuration of the training set, classification model and classification result that was deployed at any time.

c) Extensibility: should be considered as novel machine learning approaches and especially new variants of neural networks might be developed in the future.

d) Scalability: should be ensured as the classifier is intended to be used in large high-speed networks such as large corporate networks or in Internet Service Provider (ISP) environments.

e) Compatibility: is important as IDS systems can consist of multiple specialized components that for example focus on specific input data and/or attack types. Therefore, our concept should complement well with existing solutions.

C. Design

In this section, we describe the components and their interaction in a process model as well as our intended evaluation with early results.

a) Components: Our novel design consists of five main components:

System configuration journal The journal of models and system configurations: This component is responsible to track the configuration changes that are applied in each stage of our reconfiguration process. It also takes input from the human operator to label specific configurations, in case performed unsatisfactory.

Raw DGA storage The raw DGA storage, contains all (raw) DGA algorithms and blacklists on a per day and per botnet

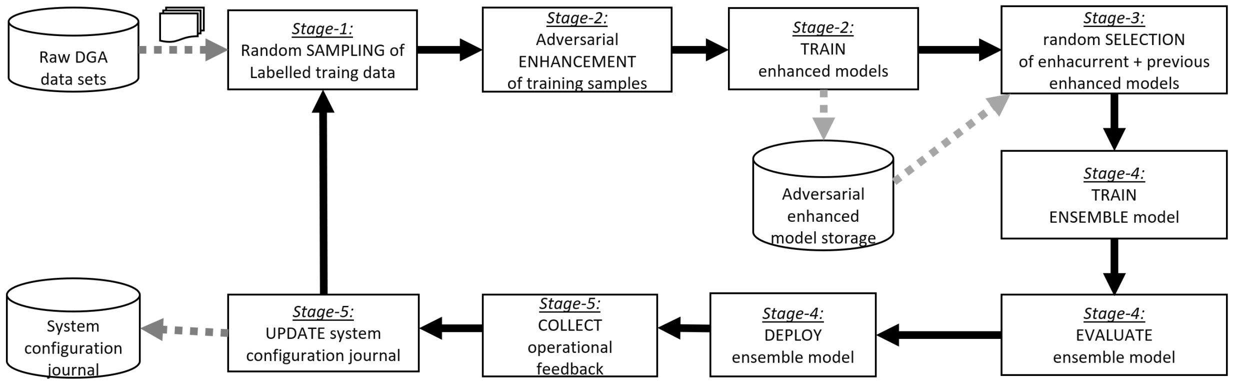
Raw DGA data sets
Stage-1: Random SAMPLING of Labelled traing data
Stage-2: Adversarial ENHANCEMENT of training samples
Stage-2: TRAIN enhanced models
Stage-3: random SELECTION of enhacurrent + previous enhanced models
Adversarial enhanced model storage
Stage-4: TRAIN ENSEMBLE model
Stage-4: EVALUATE ensemble model
Stage-4: DEPLOY ensemble model
Stage-5: COLLECT operational feedback
Stage-5: UPDATE system configuration journal
System configuration journal

Fig. 1. Process model.

basis. This component is fed by either the human operator or by subscribing to external feeds, such as [11].

Adversarial enhanced model storage This storage is used in stage-2 of our process model to store newly generated adversarial enhanced models and in stage-3 to randomly retrieve previous enhanced models. For future improvement, we designed this storage to handle models with a time-to-live (TTL). However, further research is necessary to define reasonable TTL values.

Adversarial enhancer The Adversarial enhancer component is responsible to use adversarial learning approaches to extend the training dataset with adversarial tuned DGA domains. In our current prototypical implementations we are using the *Charbot* [6] and *Deep DGA* [12] approach for this stage of our process model.

Ensemble Generator: This component combines the models selected in stage-3 of our process model and thus implements stage-4 of our process model. Our early evaluation prototype is based on simple majority vote in this component however future versions are intended to use stacked neural networks. Implements stage 4 (cross reference)This component stores the configuration in the Journal

b) Process model: The overall process model consists of the following 4 reconfiguration stages to implement a moving target defense concept against the adversary.

Stage-1: Random selection of n training samples d from the set of known domain generation algorithms D, where $n \in \mathbb{N}$.

Stage-2: Derive i copies of the previous selected samples, where $i \in \mathbb{N}$.

Stage-3: Split the enhanced set of samples into i randomly drawn sets and train n classifiers c. The classifiers are added as elements to the set of C.

Stage-4: Temporal shuffling of models. Train an ensemble model of which half the amount of included classification models is randomly drawn from the journal of previous classification models and the other half is randomly drawn from the newly created classification models. This stage firstly ensures that a significant amount of classification models that have been previously used and proven to be accurate are reconsidered for the current configuration. Secondly, it ensures that the uncertainty for the attacker and the practical complexity for adversarial attacks is increased by orders of magnitude, as the attacker now has to observe and potentially interact with the classification system over a long period of time to learn and adapt to its behaviour, which, if possible at all, will be only valid for a very limited period of time. Further, it helps rendering *Frogboiling* attacks [13] ineffective, as the attacker is unable to know how much of his previous activities is captured in the current ensemble model. Each of the four stages increases the uncertainty as random configuration choices are introduced. Therefore, our approach implements a multi-level MTD strategy as the targeted classification model is reconfigured. *Stage-5:* This stage collects the human operators feedback and updates the system configuration journal.

IV. Related work

In this section we present work related to botnet detection, adversarial learning and MTD.

Botnets have been in the focus of many related work. As the focus of our work is network based detection of botnets, we mainly considered related network based IDS or botnet detection approaches. Some of these works addressed solutions for very specific bontets that were discovered and analysed [14]. Other approaches such as [15], aim to be more generally applicable and thus being potentially applicable for yet unknown botnets or derivatives of existing one.

In the past, many IDS relied on the use of blacklists, against which DGAs were designed. The work of [16] evaluated the quality and effectiveness of such blacklists. Further research that focussed on DGAs [17] and to archive DGA domains [2] to allow attribution to specific botnet families has been done for future investigations. Phoenix [18] allows DGA based botnet tracking and intelligence. These sources provide ground truth data for training and evaluation of detection approaches.

Many related work, use machine learning as this allows to easily retrain the classifier in case new botnets appear and such models usually are trained to generalize so that similar threats e.g. new and yet unknown derivatives of the same malware family can be detected. In [19] a clustering based botnet detection approach, called Botminer, has been presented. The strength of this approach is that it operates protocol- and structure-independent. However, it has not been designed to be resilient against adversarial modified input. In [20] Disclosure, attempts to detect botnet C&C servers through large-scale netflow analysis. In relation to that the authors proposed Exposure [21], which focusses especially on DNS based detection by analysing passive DNS. Both approaches have been evaluated on large datasets and have proven to achieve high detection rates. However, neither of the two was designed with an smart attacker in mind that performs adversarial learning attacks to circumvent the systems.

One of the first approaches that was based on deep learning for DGAs was presented in [22]. This approach is based on LSTM cell based neural networks. As this work provided a reference implementation, it can be easily used as a reference implementation for our approach and serve as a benchmark. Recently, more approaches have been based on the use of deep learning. The authors of [12] proposed a method that focuses on the use of deep neural networks to improve DGA detection. Especially, generative adversarial networks (GANs) [4] are used to construct a deep learning based DGA that is able to bypass a deep learning based detector. This approach uses a series of adversarial rounds, in which the generator learns to generate domain names that are increasingly more difficult to detect. The detector model then iteratively updates its parameters to increase its coverage to the adversarial generated domains and thus becomes more resilient against adversarial attacks. This approach closely relates to stage 1 of our approach as it enhances the training samples for the classifier that is trained for detection, but does not cover the whole classifier life-cycle. The use of machine learning and especially deep learning is becoming more popular on the attackers side, which is commonly referred to as adversarial learning. Usually, adversarial learning the goal is to learn a model, e.g. a neural network, that is able generate a modified input that is able to trick another existing classification system. Thus, these systems are usually referred to as generative adversarial networks (GAN) [4]. As a consequence, these a generate a need for adversarial resilience of detection systems.

In [7] an adversarial learning technique that is called MaskDGA, which adds perturbation to the character-level representation of algorithmically generated domain names to evade DGA classifiers was presented. This approach is claimed to be applicable without requiring knowledge about the architecture and parameters of a DGA classifier. We integrate this approach into our concept as it can be used to generate 'fake' DGA samples in the first stage of our concept (presented in Section III-C0b). It is similar to an approach, called CharBot [6], which implements perturbations at character level. An enhancing framework for botnet detection using generative adversarial networks, called Bot-GEN, has been proposed in [5]. This work focusses on the discriminator rather than the generator of the GAN. The Bot-GEN framework aims to extend Netflow-based training sets by using generative adversarial networks, which continuously generate 'fake' samples to assist the original model for botnet detection and classification [5]. Similar to this approach our novel concept also extends the original training set, by adding adversarial tuned samples. However, it only addresses the training phase of machine learning based botnet detection approaches and does not take care of the whole training and deployment life-cycle management of such classification models. Bot-GAN, focusses only on Netflow-based systems, while our work currently focuses on domain dames and especially DGAs, but can be flexibly extended to other types of IDS and data sets. A key element of our novel design is the use of a multi-level MTD strategy.

The use of MTD is especially useful to hinder the reconnaissance phase of attacks where a smart attacker tries to learn the internal structure of a system, e.g. a corporate netowrk or a classifier. However, only few approaches implementing MTD in network environments exist. Previous work [23] showed that Network MTD approaches can be applied in ISP and large-scale corporate environments for other types of attacks such as Distributed Denial of Service (DDoS). Furthermore, the related work presented in [24] and [25], describe how MTD can be used to disrupt stealthy botnets.

V. Summary and Conclusions

In this paper, we described our idea and concept to make existing and future DNS based botnet detection approaches resilient against adversarial learning based attacks. We based our concept on a multi-level MTD strategy and integrated GAN based approaches. Our concept is based on a four stage (re-)training process, where we introduce random reconfigurations in each stage. This reconfiguration is intended to be executed regularly and thus limits the time for reconnaissance for the attacker for each specific configuration state of the overall detection system. This highly increases the uncertainty at the attacker side and limits the 'window of opportunity' in case of an successfully attack. In order to keep track of the already used configuration states and their accuracy, our concept includes a 'journal of system configurations'. This also allows human network operators to provide their feedback or label system states that did not perform well. Such configurations will not be used again. Our concept complements existing approaches as it is almost model agnostic. It just assumes a blackbox classifier and can be seen as a wrapper around existing solutions. It adversarially enhances the training sets and keeps track of all classifier configuration states.

For further work, we intend extended evaluation of our approach. We are currently, working on integrating and improving the MaskDGA [7] and CharBot [6] approaches in our stage-2 (see Section III-C0b), which is used for adversarial training dataset enhancement. For early evaluation work we used our approach with the LSTM based detection approach presented in [22].

Acknowledgment

This work, was partially supported by the EU project CONCORDIA (grand agreement No. 830927).

REFERENCES

[1] Cisco, "Cisco AI Network Analytics." [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/nb-06-ai-nw-analytics-wp-cte-en.html

[2] D. Plohmann, "DGArchive," 2018.

[3] A. Calleja, J. Tapiador, and J. Caballero, "The MalSource Dataset: Quantifying Complexity and Code Reuse in Malware Development," *IEEE Transactions on Information Forensics and Security*, 2018.

[4] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 1857–1865.

[5] C. Yin, Y. Zhu, S. Liu, J. Fei, and H. Zhang, "An enhancing framework for botnet detection using generative adversarial networks," in *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2018, pp. 228–234.

[6] J. Peck, C. Nie, R. Sivaguru, C. Grumer, F. Olumofin, B. Yu, A. Nascimento, and M. de Cock, "CharBot: A Simple and Effective Method for Evading DGA Classifiers," *arXiv preprint arXiv:1905.01078 [Titel anhand dieser ArXiv-ID in Citavi-Projekt übernehmen]*, 2019.

[7] L. Sidi, A. Nadler, and A. Shabtai, "MaskDGA: A Black-box Evasion Technique Against DGA Classifiers and Adversarial Defenses," *arXiv preprint arXiv:1902.08909 [In Citavi anzeigen]*, 2019.

[8] L.-F. Pau, "Botnet economics and devising defence schemes from attackers own reward processes," *arXiv preprint arXiv:1309.0522 [Titel anhand dieser ArXiv-ID in Citavi-Projekt übernehmen]*, 2013.

[9] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *2010 First IEEE International Conference on Smart Grid Communications*, 2010, pp. 350–355.

[10] D. Grochocki, J. H. Huh, R. Berthier, R. Bobba, W. H. Sanders, A. A. Cárdenas, and J. G. Jetcheva, "AMI threats, intrusion detection requirements and deployment recommendations," in *2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm)*, 2012, pp. 395–400.

[11] John Bambenek, "DGA Feed." [Online]. Available: https://osint.bambenekconsulting.com/feeds/dga-feed.txt

[12] H. S. Anderson, J. Woodbridge, and B. Filar, "DeepDGA," in *Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security - ALSec '16*, D. M. Freeman, A. Mitrokotsa, and A. Sinha, Eds. New York, New York, USA: ACM Press, 2016, pp. 13–21.

[13] E. Chan-Tin, V. Heorhiadi, N. Hopper, and Y. Kim, "The frog-boiling attack: Limitations of secure network coordinate systems," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 3, p. 27, 2011.

[14] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly resilient peer-to-peer botnets are here: An analysis of gameover zeus," in *2013 8th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, 2013, pp. 116–123.

[15] C. J. Dietrich, C. Rossow, and N. Pohlmann, "CoCoSpot: Clustering and recognizing botnet command and control channels using traffic analysis," *Computer Networks*, vol. 57, no. 2, pp. 475–486, 2013.

[16] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *International Workshop on Recent Advances in Intrusion Detection*, 2014, pp. 1–21.

[17] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *25th 5USENIX6 Security Symposium (5USENIX6 Security 16)*, 2016, pp. 263–278.

[18] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, "Phoenix: DGA-based botnet tracking and intelligence," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, 2014, pp. 192–211.

[19] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," 2008.

[20] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, 2012, pp. 129–138.

[21] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," in *Ndss*, 2011, pp. 1–17.

[22] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," *arXiv preprint arXiv:1611.00791 [In Citavi anzeigen]*, 2016.

[23] J. Steinberger, B. Kuhnert, C. Dietz, L. Ball, A. Sperotto, H. Baier, A. Pras, and G. Dreo, "DDoS Defense using MTD and SDN," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–9.

[24] S. Venkatesan, M. Albanese, G. Cybenko, and S. Jajodia, "A moving target defense approach to disrupting stealthy botnets," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, 2016, pp. 37–46.

[25] M. Albanese, S. Jajodia, and S. Venkatesan, "Defending from stealthy botnets using moving target defenses," *IEEE Security & Privacy*, vol. 16, no. 1, pp. 92–97, 2018.